1394 Device Power Management Proposal

Version 0.7

October 9, 1996

Proposal to: **Architecture Working Group of the 1394 Trade Association**

This specification is submitted for feedback in preparation for a working group meeting. Feedback may be sent to the TA reflector **1394-sig@apple.com** or to **1394@microsoft.com**. The 1394 Device Power Management specification will be appended to the 1394 Cable Power Distribution specification and submitted as a 1394 Trade Association Specification.

Draft by Microsoft and Apple Computer

Abstract: This document specifies extensions to the current IEEE 1394-1995 specification which are necessary to implement device power management. Device power management is important to conserve power for effective use of available cable and battery power and improved reliability. The device power management architecture assumes the power distribution model specified in the 1394 TA Cable Power Distribution Specification. Together, these specifications define a standard model for 1394 power distribution and device power consumption for a consistent, user friendly, device response.

Table of contents

1. Overview

IEEE 1394-1995, High Performance Serial Bus, has been ratified as a standard by the IEEE balloting group. However, while the IEEE 1394-1995 specification does provide a rich set of features for implementation of cable power distribution, the standard does not provide guidance on application of these features for 1394 device power management. This document describes extensions to the standard that are necessary to manage device power consumption independent of whether the power is drawn from the cable, a local AC power supply or a battery. This specification addresses power state definitions, power state control interfaces, and power state transitions.

This specification assumes that devices implement the common power distribution and control architecture as specified in Part I of this 1394 Power Specification. Implementation of Part I, 'Cable Power Distribution Model' is a pre-requisite for a power managed environment. All power control functions are performed by the elected bus manager. No device performs asynchronous local power control except in the absence of an elected bus manager.

The goal of 1394 Device Power Management is to provide a consistent method for implementing low power 1394 devices, conserving power and maximizing device availability to create a positive experience for the end-user. The mechanisms allow a qualified Bus Manager to implement a bus-wide power management policy to deliver these benefits. Devices which implement this specification are intended to function well in both an unmanaged and a power managed environment.

1.1 Definition of Terms

1.2 Related Documents

IEEE 1394-1995 Standard, High Performance Serial Bus

2. 1394 Power States

Device power management refers to the manipulation of power states and capabilities on a per-function (i.e. per-1394 Unit) basis. For convenience, a generalized model is defined consisting of 4 power states. This gives Units flexibility in exposing appropriate power states, while giving the managing entity a reasonable number of states to control. For the purposes of device power management on 1394, Nodes also have defined power states. Node power states enable an additional level of control that the Bus manager can use to optimize node power consumption once unit power consumption has been minimized by the managing entity.

A 1394 node can be power managed at the node level, all the way down to a per Unit granularity. Control of power at the node level is done by the bus manager. The bus manager can enforce node power control by examining the Self Id packets, and determining if enough power is available to satisfy the power requirements found in the Self Id packet of a particular node before turning it's link on. Since the bus manager will not have any knowledge of the usage of a node's unit, it will not attempt to power manage these individual units. If a unit requires more power than what was expressed in the nodes power requirements field of the Self Id packet, the unit must express what it's additional power requirements are as discussed in Section 3.2. Once the additional power requirements have been met, or if no additional power requirements were required by the unit, the entity managing this unit can transition the unit to a higher/lower power state.

1394 Node power states are controlled by the Bus Manager using mechanisms specified in the defining IEEE 1394 standard such as the Link on packet and clearing the Link bit in the State_Clear register. These states relate to the basic functional states of the bus, and are included here in order to describe their use in a power managed environment. No additional definitions are needed to implement or control Node power states.

2.1 Node Power States

1394 bus interface silicon consists of a PHY bus transceiver layer and a Link packetization layer. In accordance with the 1394 TA Cable Power Distribution Specification, and to facilitate device power management, the following power states have been defined for 1394 nodes.

Node Power State	Operational State of Node	Notes
N0 (Full on)	PHY: On Link: On Unit(s): Fully-powered	-Configuration ROM accessible -CSRs accessible; preserved -Units may* consume per-unit power requirements
N1 (Link on)	PHY: On Link: On Unit(s): Low-power	-Configuration ROM accessible -CSRs accessible; preserved -Units shall not consume per-unit power requirements
N2 (PHY on)	PHY: On Link: Off Unit(s): Low-Power	-Per 1394 definition for Link off
N3 (off)	PHY: Logically off Link: Off Unit(s): Off	-Power may be removed from the PHY

1394 node power states are characterized by the operational state of the node and its power consumption, as well as the latency required for transition to N0 (full on) from another state.

N0 (Full on)

In the N0 state, the PHY and Link are fully functional and units may* be consuming incremental power as defined in their Unit Power Requirements registers. Thus the node context is fully accessible including the configuration ROM and control and status registers (CSRs). Power consumption in the N0 state is defined by the 1394 specification, and consists of less than 1Watt for the PHY, plus the Link_on power increment specified in the Power Class field of the Self_ID packet (if any), plus any Unit Power Requirements specified in the Unit Directories according to the power state of each Unit.

*Note: The N0 and N1 states may be equivalent if the node in question does not have any additional per unit power requirements.

N1 (Link on)

Per the 1394 specification, Link On must enable full access to the configuration ROM and Control and Status Registers (CSRs). In the N1 state the bus manager has access to Unit directories to determine the Unit power capabilities, implemented Unit power states, and associated Unit Power Requirements. The latency to return the node to N0 from N1 is less than that to return to N0 from N2. Power consumption in the N1 state consists of PHY power (< 1 Watt) plus the Link_on power increment specified in the Power Class field of the Self_ID packet (if any). While at the N1 state, the node can power whatever units are necessary, as long as they require no more additional power than what was specified by the Self_ID packet.

N2 (PHY on)

This state is defined by the 1394 specification for the Link off condition. When the device's logical link context is disabled in transition to the N2 state, a bus reset is required to notify other nodes that the node's node_id has gone away. The latency to return the node to N0 from N2 is less than that to return to N0 from N3, but can be more than to return to N0 from N1. In the N2 state, node power consumption is less than 1 Watt as defined for the PHY, and node functions are limited to bus repeater and bus reset.

When a node is not being used, it may be put into the N2 state yet still maintain the electrical integrity of the bus.

N3 (Off)

N3 is the lowest power state of the node. In the N3 state, the PHY power may be removed and the logical context of the node and it's PHY bus repeater function may be disabled. State N3 is entered only by the device in response to the Power State Change Request/Response Protocol (sec. 3.8) or by the user physically removing a node's power source.

2.2 Unit Power States

Each Unit on a 1394 node implements some number of Unit power states appropriate for that Unit. Unit power state D0 (Full-on) is required and is implemented by default. Additional low-power states (D1-D3) may be implemented according to class-specific needs, and in compliance with any Class-specific Power Management Specifications that apply. For each implemented Unit power state, any necessary incremental Unit Power Requirements for that state are listed in the Unit Directory as described in section 3.1.1. Because of the requirement for Link_On in order to access the Unit directory, Unit power states are only visible and controllable in node power states N1 and N0. Unit power states cannot change while the node is in N2 or N3, and units must maintain their power states during this time. The bus manager must ensure that node power state is always logically higher than or equal to all Unit power states on the node.

Unit Power State	Operational State of Unit	Notes
D0 (Full on)	Fully functional	
D1	Class-specific	Power consumption: <D0; >D2 Latency to D0: <D2 Unit register context can be preserved by Unit
D2	Class-specific	Power consumption: <D1;>D3 Latency to D0: <D3; >D1 Unit register context may be lost by Unit
D3 (Off)	Logically Off	Power may be removed from the Unit

2.3 Impact of bus reset on Power States

When a node is hot plugged into the 1394 bus, power is applied to its PHY placing it in the N2 power state while also triggering a bus reset. Among other things, this reset is used by the bus manager to make the necessary power checks and to reconfigure cable power distribution, if necessary. If the cable power source drops below some critical threshold at a particular node, then the node in question should issue a bus reset so all nodes can revert back to their N2 state, as required by the 1394 specification. In all other cases, however, the bus reset simply postpones bus transactions and has no affect on the power state of nodes or units.

2.4 Soft Power-On Protocol

A 1394 Unit is brought to full functionality in two stages. The first stage is that the node is transitioned from the N2 state to the N1 state by the bus manager. Once the node is in the N1 state, another controlling entity can power up it's individual units as needed. The unit's possible power states are described by the Unit Power Capabilities entry located in its Unit directory entry (see section 3.1). The unit is placed into higher power states, (if implemented), by modifying the Power State Control register (see section 3.2).. After modifying the Power State Control register, the entity wishing to power up a particular unit may need to allocate additional power as specified by the Unit Power Requirements register(s) (see section 3.1.1). By querying the appropriate Unit Power Requirements register, the entity can allocate additional power from the bus manager at the NODE_POWER_AVAILABLE register (see section 4.2).

3. Device Power Management Interface

This section describes the standard mechanisms for controlling the power states of a 1394 Unit. The necessary functions consist of the following:

- Mechanism for describing Unit power capabilities.
- Control for setting Unit power states.
- Mechanism for reporting Battery Unit status.
- Protocol for a node to wake the Bus Manager (or other node) upon a pre-defined event
- Protocol for nodes to signal the Bus Manager when power-related events occur

The result is a user friendly power managed environment that conserves power, monitors status of battery-powered devices and provides for low latency power-on of devices.

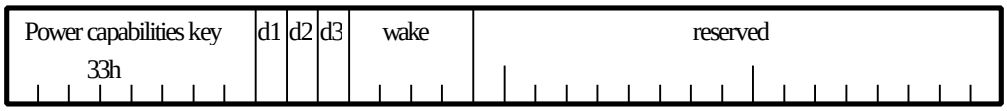
Devices that implement the Device Power Management Interface must set the pme bit in the Bus_Info Block (see Appendix A for example Configuration ROM).

As with all 1394 CSRs, reserved bits must return zero on reads, and ignore writes.

3.1 Power_Capabilities entry

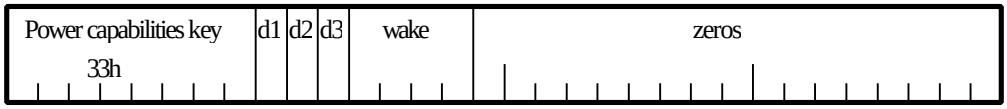
Units must describe their power-related capabilities in order to participate in the bus-wide power management policy. The Power_Capabilities entry shall be implemented as a read only entry within a given Unit Directory.

Definition

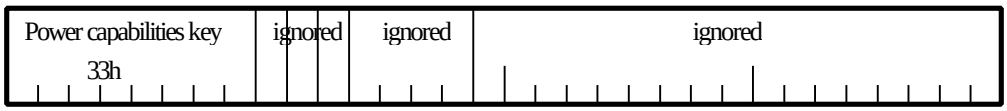


initial values

read values



write effect



- d1: Indicates whether unit implements power state d1 (1 bit)
- d2: Indicates whether unit implements power state d2 (1 bit)
- d3: Indicates whether unit implements power state d3 (1 bit)
- wake: Bitmask indicating which power states unit can signal wakeup from (4 bits)

  0001b = can generate wake up from D0 state

  0010b = can generate wake up from D1 state

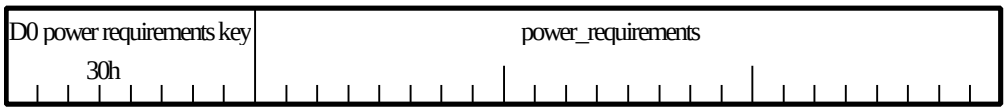
0100b = can generate wake up from D2 state

1000b = can generate wake up from D3 state

3.2 Unit_Power_Requirements entry

If a unit consumes bus power in excess of the level specified by pwr field in the Self_ID packet transmitted by the node, then the unit must specify a Unit_Power_Requirements entry within it's Unit Directory. This Unit_Power_Requirements entry specifies how much additional power is necessary in order for the unit to become functional at the given Dn state. The 24 bit power_requirements field specifies the additional bus power requirements of the unit, in deciwatts, needed to become functional for the given Dn state.

entry for D0 state



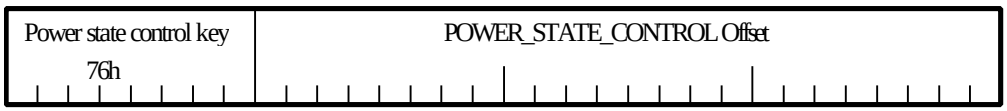
entry for D1 state

entry for D2 state

3.3 Power_State_Control entry

The Power_State_Control entry is implemented within a Unit Directory, and serves as a pointer to the actual POWER_STATE_CONTROL register.

definition

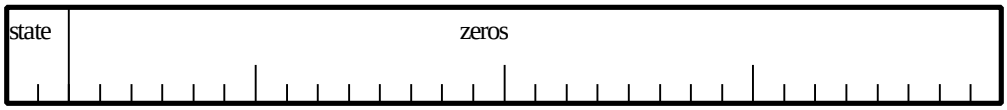


3.4 POWER_STATE_CONTROL register

The POWER_STATE_CONTROL register reports/controls what Dn state the unit is in at any given time. This register should be implemented as a compare_swap only register to avoid race conditions that may occur when allocating/releasing additional power for this unit. If additional power is required to transition this unit to a higher Dn state, such power must be expressed by a Unit_Power_Requirements entry (see Section 3.2). Only after a successful compare_swap operation has occurred, should additional power be requested/released from the bus manager at the NODE_POWER_AVAILABLE registers (see Section 4.2)

initial values

read values



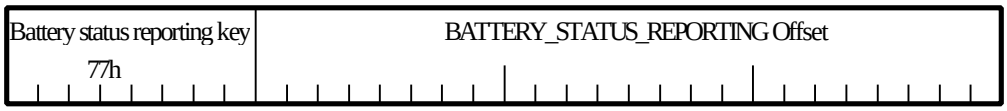
write effect

- state: field indicating the unit's current power state (2 bits)
  00b = D0 state
  01b = D1 state
  10b = D2 state
  11b = D3 state

3.5 Battery_Status_Reporting entry

Nodes which contain a battery may report its existence by including a Battery Unit Directory in the nodes Configuraton Rom. Multiple Battery Unit Directories are possible. Each Battery Unit Directory contains at least a Battery_Status_Reporting entry. The Battery_Status_Reporting entry is implemented within a special Battery Unit Directory, and serves as a pointer to the actual BATTERY_STATUS_REPORTING register.

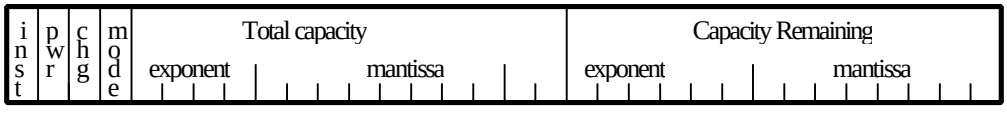
definition



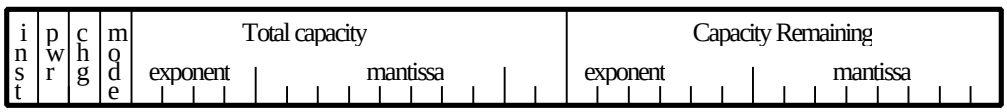
3.6 BATTERY_STATUS_REPORTING register

The BATTERY_STATUS_REPORTING register allows the status of a battery to be queried.

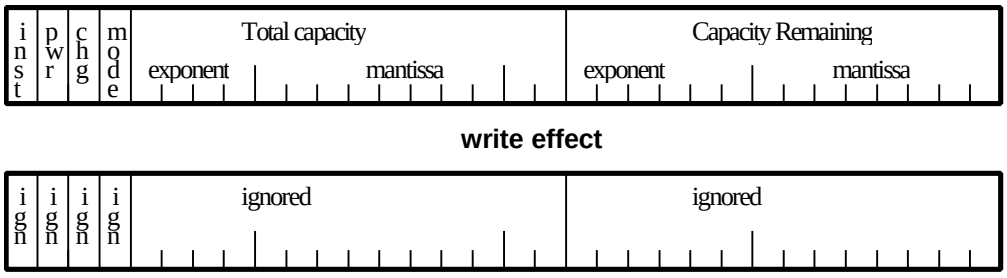
definition



initial values



read values



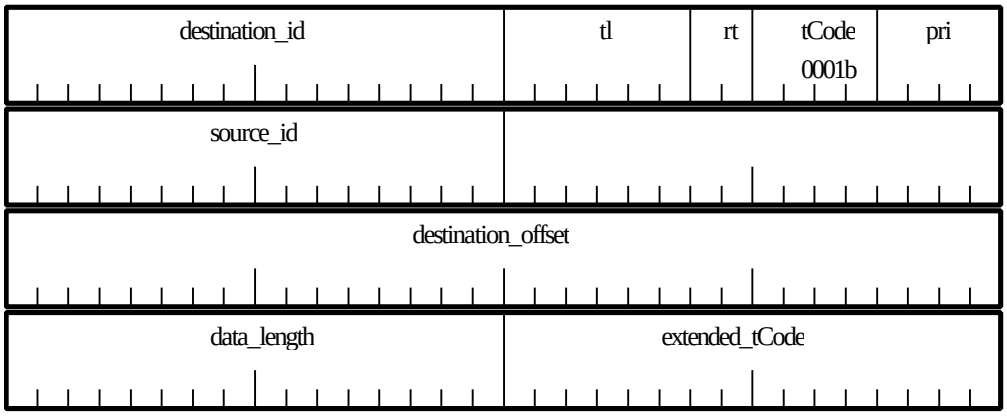
- inst = the battery is currently installed (1 bit)
- pwr = the battery is currently powering the node (1 bit)
- chg = the battery is currently charging (1 bit)
- mode = capacity mode (0=milliamp-hours, 1=milliwatt-hours) (1 bit)
- Total capacity expressed in milliamp-hours or milliwatt-hours (14 bits). This field is expressed as a 10 bit mantissa and a 4 bit exponent.
- Capacity remaining expressed in milliamp-hours or milliwatt-hours (14 bits). This field is expressed as a 10 bit mantissa and a 4 bit exponent.

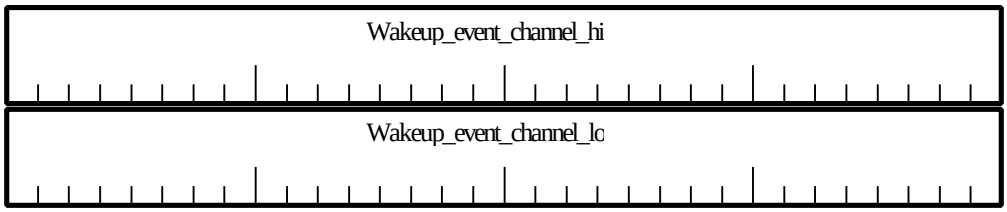
3.7 Wakeup Event Notification

It will sometimes be desirable for a 1394 device to wake up a host CPU in a low power mode. For instance, a sleeping CPU may wish to be woken up when a 1394 fax-modem receives an incoming call. While in a sleeping state, the CPU's ability to communicate on the 1394 bus may be very limited. In the case of portables, the CPU may not be able to run driver software. In this case, the 1394 adapter hardware may be required to support the wakeup mechanism.

Thus, a simple, standard wakeup mechanism is defined as a part of the CSR register set. Devices supporting wakeup events send a broadcast packet to a fixed CSR register address. Devices implementing this address check the packet payload to determine if it indicates that the receiving device should wake up. A simple bit test is all that is required to make this determination. The wakeup event notification packet is defined as follows:

Wakeup Event Notification Broadcast Packet



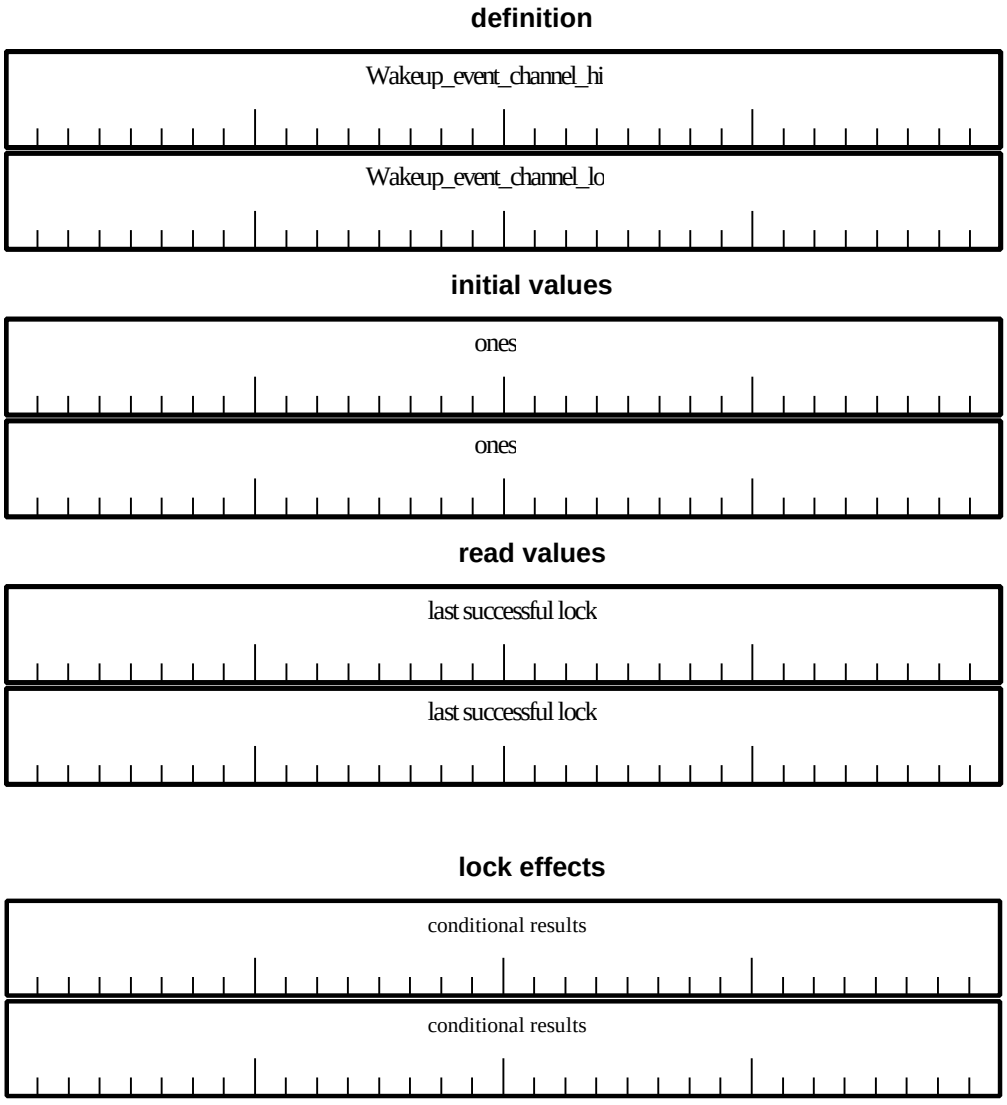


Each device desiring to receive wakeup notification listens to a single wakeup event channel. Assignment of event channels is defined below. When a device receives a wakeup event notification packet with its event channel bit set, it wakes up. After wakeup, higher level software will be able to determine the source of the event from the source ID of the packet. Further determination of the actual event may be done by communicating with the notifying device.

Each device that can send wakeup event notifications shall implement a 64-bit wakeup event channel register for each type of event it supports. A device wishing to be woken up will set its wakeup event channel bit in the wakeup event channel register corresponding to the event it wants to be woken up on. When the notifying device detects that a particular event has occurred, it broadcasts the wakeup event packet with a payload equal to the contents of the wakeup event channel register corresponding to the detected event.

Assignment of event channels is done in a fashion similar to allocating isochronous channel numbers. The power management node shall implement a 64 bit CSR wakeup event channels available register defined as follows:

WAKEUP_EVENT_CHANNELS_AVAILABLE register



Devices wishing to allocate a wakeup notification event channel use the compare and swap lock transaction to clear one of the available bits. If this operation is successful, the allocated bit is the bit the device should use to be woken up.

3.8 Power Status Change Notification/Request Protocol

In order to prevent bus integrity or device functional failures due to unexpected device node power changes, a power status change notification/request protocol is needed. When a device node desires to change it's power status in a way that may affect the bus integrity or functionality of devices, it must first request permission to do so from the bus power manager. The bus power manager will check if the bus power requirements can still be satisfied when the device's power status changes, and will also check with unit-specific software to ensure that device functionality is able to be cleanly closed-down. If these checks are successful, the bus power manager will grant permission to the device to change it's status. If the checks are not successful, the bus power manager will attempt to reduce the bus power requirements and/or will allow the unit-specific software to preserve user data and cleanly close-down, interacting with

applications and the user as necessary. Once this is done, the bus power manager will grant permission to the device node to change it's status.

For the case when a device node is running off of a battery that is about to lose power, the device will notify the bus power manager. The battery powered device should, preferably, wait until being granted permission to stop providing power, as described above. However, since the remaining power is limited, the battery powered node may remove power without permission.

The power state change notification/request protocol registers are defined below:

3.9 POWER_STATUS_CHANGE_NOTIFY_REQUEST register

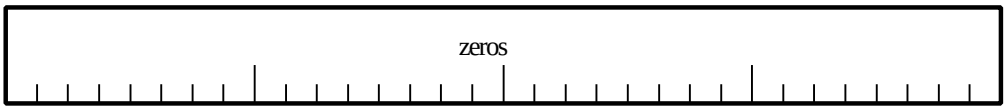
Definition

PR =1: User Power-off Request

FN = 1: Imminent Power Fail Notification

CN = 1: Power Source Change Notification

initial values

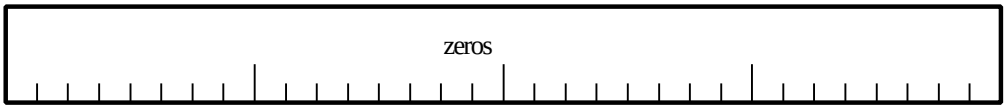


write effects

* = 1: Processes Request

** = 1: Processes Notification

read values



3.10 POWER_STATE_CHANGE_REQUEST_RESPONSE register

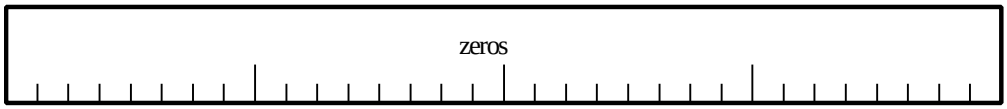
Definition

GP = 1: Grant Pending

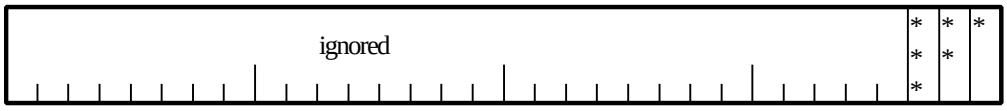
RG = 1: Request Granted

RD = 1: Request Denied

initial values



write values



* = 1: Wait for Grant

** = 1: Change Status

*** = 1: Do not Change Status

read values

When a power status change event occurs on a device, the device writes a 1 into the appropriate bit of the POWER_STATE_CHANGE_NOTIFY_REQUEST register of the bus power manager. The device waits for 1 second for a response from the bus power manager. If the power provider does not receive a response, it may change it's status. The bus power manager responds to a status change request by writing to the power provider's POWER_STATE_CHANGE_REQUEST_RESPONSE register.

The bus power manager, along with unit-specific software, will try to ensure that a bus power failure or data loss will not occur when the device changes power status. It must first determine if the bus power requirements must be reduced. If they must be reduced, the bus power manager must attempt to reduce

them. If the bus manager determines that a bus power failure or data loss will not occur, it may grant permission by writing a 1 to the request_granted bit. If the bus manager determines that a bus power failure or data loss will occur, it may deny permission by writing a 1 to the request_denied bit. Note that the device is not obligated to change status even when it has been granted the ability to do so.

If the bus power manager cannot grant or deny permission within 1 second, it may respond with a pending grant by writing a 1 to the grant_pending bit. The bus power manager must then respond with a grant, denial, or grant pending within 1 second; otherwise, the device may change power status.

In the case that the device is battery powered, it should write a 1 to the Imminent Power Fail Notification bit of the POWER_STATE_CHANGE_NOTIFY_REQUEST register. It should attempt to follow the protocol for non battery powered devices, but may stop providing power at any time after sending the notification.

3.11 Bus Manager Override

The 1394 bus power management design specifies additional tasks for the 1394 bus manager to perform in order to improve bus power management. The design may be enhanced in the future, requiring more tasks for the 1394 bus manager. It may sometimes be the case that multiple bus manager capable nodes may be present on the same bus. These nodes, however, may implement different versions of bus power management. Thus, a method is required for determining the node with the latest capabilities and setting that node as the bus manager.

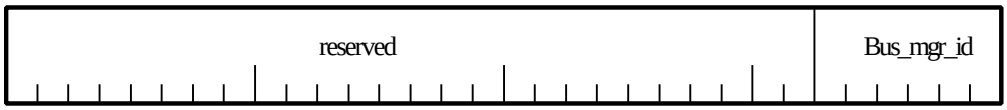
The first requirement is for all bus management nodes to provide bus management version information. This will reside in the bus info block in the reserved bit fields in the second quadlet as follows:

Configuration ROM
(located at FFFF F000 0400 $_{16}$)

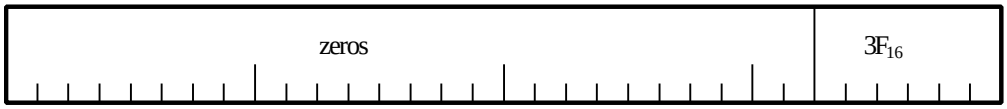
| Block | Offset | Description | | |

3.12 BUS_MANAGER_OVERRIDE register

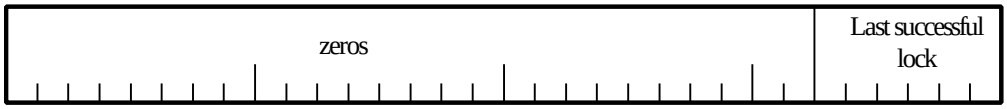
Definition



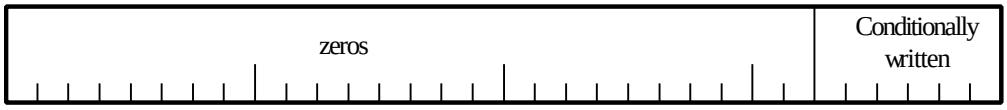
initial values



read values



lock effects



When the bus manager override register is successfully updated on the incumbent bus manager node, that node will relinquish its incumbency to the overriding node (the node that updated the bus manager override register) and initiate a bus reset. The overriding node will then have 125 milliseconds to establish itself as the new bus manager. Since the original bus manager node relinquished its incumbency, it will wait 125 milliseconds along with the other challengers.

To establish the most capable bus management node as the incumbent bus manager, the following algorithm will be used. First, normal bus management contention will be used to establish a bus manager. This bus manager may or may not be the most capable bus management node. Once a bus manager is established, the other challengers will examine the incumbent's bus management version info in the bus info block. If a bus manager challenger is more capable (greater version number), it will attempt to override the incumbent bus manager using the bus manager override register.

If a bus management override occurs, the above process is continued.

Additional overriding may take place until the node with the highest bus manager version is the incumbent bus manager. At this point, no other challenger should attempt an override.

4. Bus Power Management Function

The basic 1394 power distribution model is specified in the: 1394 TA Specification for Cable Power Distribution. The bus power management functions described here are consistent with this specification. This section describes bus power management protocols.

4.1 Overview of Power-on initialization with a PC Bus Manager

Bus Initialization

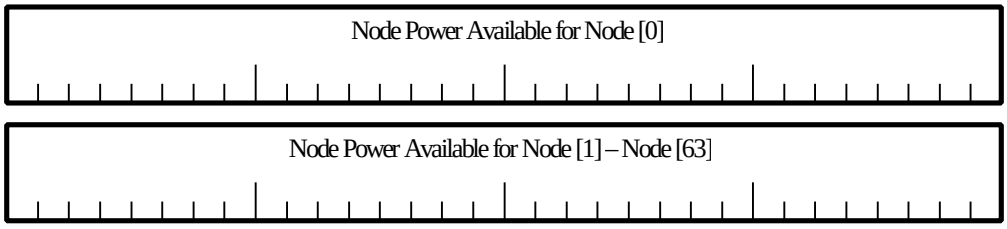
The 1394 bus is configured in response to a bus reset immediately following a power on, or a device hot plug or unplug. Each device node is in the N2 node power state drawing < 1Watt during power on initialization.

During the PC's power up initialization, it discovers and initializes a 1394 host controller. In order to discover what nodes are present on the bus, the PC resets the bus as part of its normal initialization process.

If the PC determines that it is the bus manager, it will try to power up nodes permitting enough power is available. In order to facilitate varying ranges of Bus Manager capabilities, a new register will be utilized to override the bus manager. This BUS_MANAGER_OVERRIDE will be discussed in an addendum to the IEEE 1394 specification.

4.2 NODE_POWER_AVAILABLE registers

The bus manager shall maintain the amount of available bus power that each node on the bus can allocate and use. This information is located at a bus manager who has identified himself as a power management capable bus manager (as defined by power management bit turned on within the bus information block). These registers shall be located at a defined location within the bus managers CSR space. Figure 1-1 shows the format of these NODE_POWER_AVAILABLE registers.



There are 64 NODE_POWER_AVAILABLE registers which correspond to every node found on the bus. A node wishing to use more bus power than what it reported in it's Self ID packet should use its own physical id as an index into the NODE_POWER_AVAILABLE registers and attempt to do a compare swap lock transaction in order to secure the amount of power it requires. If the node's allocation for more bus power was successful, the bus manager will adjust any other NODE_POWER_AVAILABLE registers necessary in order to reflect the amount of remaining bus power available for that particular node. A node wishing to release the additional power it previously allocated, must use a corresponding compare swap lock transaction to its entry within the NODE_POWER_AVAILABLE registers to return the allocated bus power back to the available pool. Power allocation/deallocation is performed by a Unit only in conjunction with Unit power state changes (see Section 3.4).

The NODE_POWER_AVAILABLE registers are 32 bits wide and specified in deciwatts, and uses the same format as the Unit_Power_Requirements entry (see Section 3.2). Although the

NODE_POWER_AVAILABLE registers have 8 more bits than the Unit_Power_Requirements, it is expected that only the lower 24 bits are meaningful.

As described in section 3.2, any unit that needs more bus power than that reported in it's Self ID packet, must report this information in its unit directory using the Unit_Power_Requirements entry for any implemented Unit power states.

4.3 Soft Power Shutdown and Device Removal

When the power switch on a node is pressed, the device will notify the bus using the Power Status Change Notification/Request protocol. Battery powered devices with their own power cord may be self powered or source power to the bus. On removal of the AC power cord, the device must use the Power Status Change Notification/Request protocol to notify the bus manager of a power source change.

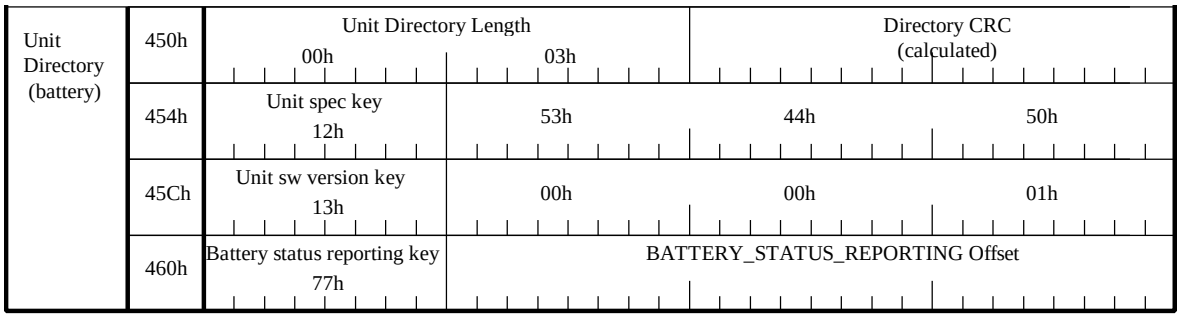
To facilitate emergency power shutdown under power fail conditions, devices sourcing cable power are required to support the Power_Fail_Imminent and Power_Source registers specified in clauses 8.3.2.3.3 and 8.3.2.3.4 of the IEEE 1394-1995 standard.

5. Appendix A: Example Configuration ROM

This section provides an example configuration rom that combines several elements of this specification outined above. The node in question informs others on the bus that it is a power management enabled node, by turning on the new pme bit within the Bus_Info_Block @ offset 408h. The example below implements the power capabilities entry (key = 33h), the power state control entry (key = 76h), as well as two additional power requirement entries (key = 30h & 31h) all within the unit directory. This node also utilizes batteries to power the node. Thus a battery unit directory was required within this configuration rom. That unit spec key of 53h 44h 50h identifies this unit as a battery unit. At the very least this unit must implement the battery status reporting key to serve as a pointer to the BATTERY_STATUS_REPORTING register.

Configuration ROM
(located at FFFF F000 0400_{16})

Block	Offset	Description																															
First Quadlet	400h	bus_info_length 04h															CRC_length 17h															ROM_CRC_value (calculated)										
Bus_Info Block	404h	'1394' in ASCII 31h															33h															39h										34h
	408h	irmc		cmc		isc		bmc		pme		reserved												cyc_clk_acc																max_rec								reserved										
	40Ch	node_vendor_id																																									chip_id-hi										
	410h	chip_id-low																																																		
Root Directory	414h	Directory Length 00h															04h															Directory CRC (calculated)										
	418h	vendor ID key 03h															module_vendor_id																																						
	41Ch	node capabilities key 0Ch															node_capabilities 00h															83h										80h										
	420h	node unique id key 8Dh															node_unique_id leaf offset 00h															00h										02h										
	424h	unit directory key D1h															unit directory offset 00h															00h										04h										
Node Unique ID Leaf	428h	length of leaf 00h															02h															Leaf CRC (calculated)										
	42Ch	node_vendor_id																																																		chip_id_hi										
	430h	chip_id_lo																																																		
Unit Directory	434h	Unit Directory Length 00h															06h															Directory CRC (calculated)										
	438h	Unit spec key 12h															unit_spec_id																																						
	43Ch	Unit sw version key 13h															unit_sw_version 01h															04h										83h										
	440h	Power capabilities key 33h															d1		d2		d3		wake									reserved																						
	444h	Power state control key 76h															POWER_STATE_CONTROL Offset																																						
	448h	D0 power requirements key 30h															power requirements																																						
	44Ch	D1 power requirements key 31h															power requirements																																						



6. Appendix B: Cable-Powered Bus Manager Initialization

The following diagram depicts what steps are required for a bus manager which needs to draw cable power. In an ideal power managed environment, this bus manager node should initially be in the N2 node power state (phy on only). This node in question should inspect the Self_ID packets it received from the bus reset upon initially joining the bus. From these Self_ID packets, the node should determine whether or not another node exists that does not require cable power and is bus manager capable. If another such node exists, then this node does not have to perform any special action. However, if another such node does not exist, this node must follow the steps outlined below in order to Assert himself as the bus manager. An example of this scenario would be where a portable is capable of being bus manager, but it is powered by power brick.